Introduction

T. H. Pulliam

Intro to Computational Fluid Dynamics (CFD)

Instructor: Dr. Thomas H. Pulliam, (650)-604-6417

email: thomas.h.pulliam@nasa.gov

snailmail: MS 258-2 NASA Ames Research Center, Moffett Field, CA, 94035

Prerequisites: Undergraduate course in Fluid Mechanics and Thermodynamics, Compressible flow, Linear Algebra (or consent of instructor).

Notes/Textbook: Fundamentals of Computational Fluid Dynamics by Lomax, Pulliam and Zingg. Published by Springer-Verlag, ISBN=3-540-41607-2 Available at Bookstore or any book website

Course Summary: To develop an understanding of Computational Fluid Mechanics and provide an opportunity to practice numerical solution techniques as applied to the equations governing Fluid Mechanics and Heat Transfer. The mathematical structure is the theory of linear algebra and the attendant eigenanalysis of linear systems. The course will focus on the development, analysis and use of numerical tools (as applied to stability, accuracy, and design methods based in linear theory) to develop a basic understanding of algorithms and methods of practical value (e.g. methods for the Euler and Navier-Stokes equations). Topics include, explicit and implicit time differencing methods, central, upwind and characteristic spatial differencing techniques, classical relaxation, multigrid methods and splitting/factoring methods. Practical examples and real life lessons will be shared in the hope of developing a feel for contemporary methods and codes. Extensive use of MATLAB for the problems,exercises and project.

Computational Fluid Dynamics

- Numerical Solution of Partial Differential Eqs., PDE's.
- Application to a Wide Variety of Physical Systems:
  - Aerodynamics, Fluid Mechanics, Astronautics
  - Heat Transfer, Combustion, Magnetohydrodynamics
  - Astrophysics, Weather Prediction, Ocean Modeling
- Solve the PDE on a Discrete Lattice (Grid, Mesh, Tessellation).
- Convert PDE to a System of Semi-Discrete Eqs., ODE's.
- Convert the Semi-Discrete ODE's to Fully Discrete OΔE's.
- Systematic Design, Analysis and Implementation of Methods.
- Accuracy, Consistency, Stability, Convergence, and Efficiency.

Navier-Stokes Equations

$$\partial_t Q + \partial_x E + \partial_y F = Re^{-1}\left(\partial_x E_v + \partial_y F_v\right)$$

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \quad E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(e+p) \end{bmatrix}, \quad F = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(e+p) \end{bmatrix},$$

$$E_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ f_4 \end{bmatrix}, \quad F_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ g_4 \end{bmatrix}$$

$$
\begin{aligned}
\tau_{xx} &= \mu(4u_x - 2v_y)/3 \quad \tau_{xy} = \mu(u_y + v_x) \quad \tau_{yy} = \mu(-2u_x + 4v_y)/3 \\
f_4 &= u\tau_{xx} + v\tau_{xy} + \mu Pr^{-1}(\gamma - 1)^{-1}\partial_x a^2 \\
g_4 &= u\tau_{xy} + v\tau_{yy} + \mu Pr^{-1}(\gamma - 1)^{-1}\partial_y a^2
\end{aligned}
$$

Pressure is related to the conservative flow variables, Q, by the equation of state

$$
p = (\gamma - 1)\left(e - \frac{1}{2}\rho(u^2 + v^2)\right)
$$

where γ is the ratio of specific heats, generally taken as 1.4. The speed of sound is a which for ideal fluids, $a^2 = \gamma p/\rho$. The dynamic viscosity is μ and is typically made up of a constant plus a computed turbulent eddy viscosity. The Reynolds number is Re and Prandtl number Pr.

Finite-Volume Methods

- Finite-volume methods are applied to the integral form of the governing equations.

$$\frac{d}{dt}\int_{V(t)} Q dV + \oint_{S(t)} \mathbf{n}.\mathbf{F} dS = \int_{V(t)} P dV$$

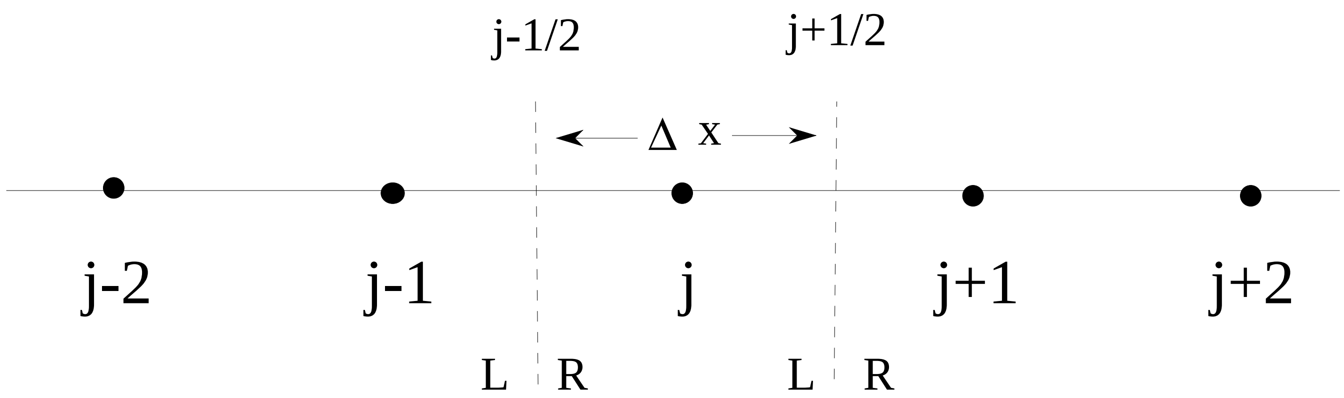


Figure 1: Control volume in one dimension.

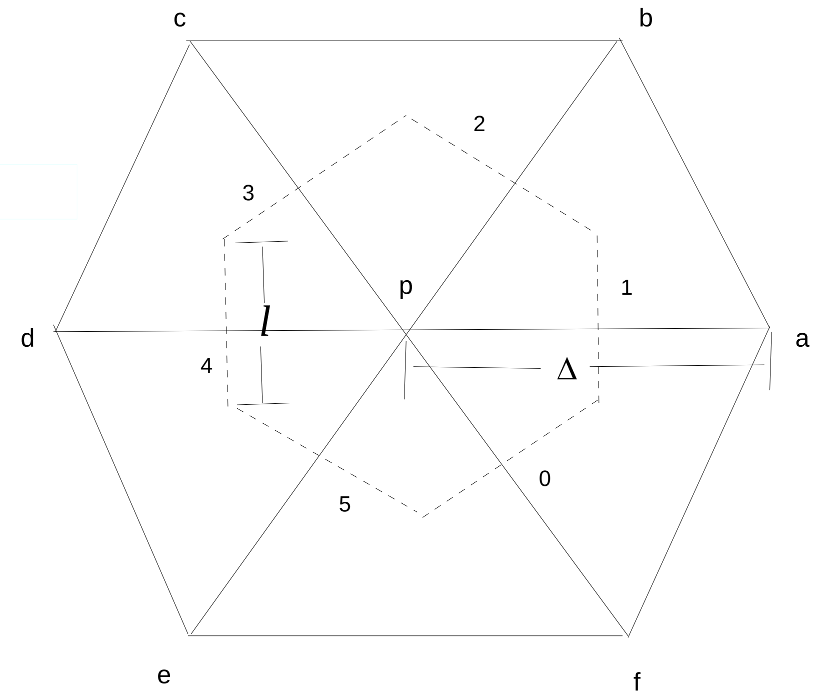


Figure 2: Control volume in Unstructured Triangular Grid.

- Unstructured primitive can be any N-side polygons.

Finite-Difference Schemes

- Approximate Equations Discretely in Space and Time

$$x = x_j = j\Delta x \qquad t = t_n = n\Delta t = nh \qquad u(x + m\Delta x, t + kh) = u_{j+m}^{(n+k)}$$

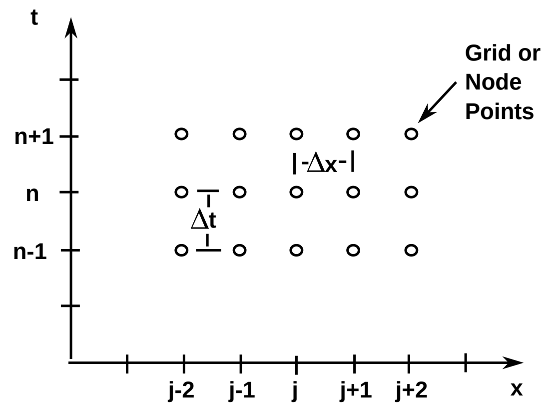


Figure 3: Space-time grid arrangement.

- Using Taylor Series Expansions

$$u_{j+1} = u_j + (\Delta x)\left(\frac{\partial u}{\partial x}\right)_j + \frac{1}{2}(\Delta x)^2\left(\frac{\partial^2 u}{\partial x^2}\right)_j \ldots + \frac{1}{n!}(\Delta x)^n\left(\frac{\partial^n u}{\partial x^n}\right)_j \ldots$$

- Rearranging we get a finite-difference approximation to the 1^{st} Derivative, with truncation error er_t

$$\begin{aligned}
\frac{u_{j+1} - u_j}{\Delta x} &= \left(\frac{\partial u}{\partial x}\right)_j + \frac{1}{2}(\Delta x)\left(\frac{\partial^2 u}{\partial x^2}\right)_j + \cdots \\
er_t &= -\frac{1}{2}(\Delta x)\left(\frac{\partial^2 u}{\partial x^2}\right)_j
\end{aligned}$$

Finite-Difference Methods

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

$$\frac{Q^{n+1}_{j,k} - Q^{n}_{j,k}}{\Delta t} + \frac{\left(E_{j+1,k} - E_{j-1,k}\right)^{n+1}}{2\Delta x} + \frac{\left(F_{j,k+1} - F_{j,k-1}\right)^{n+1}}{2\Delta y} = 0$$

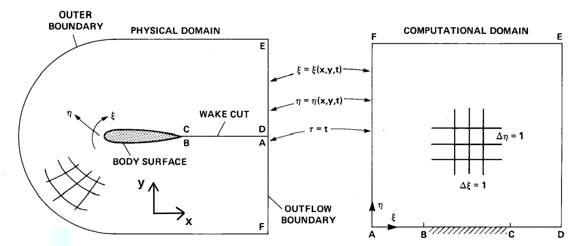


Figure 4: Physical and computational spaces.

Time Advance Schemes

- Integrate Equations Forward in Time
- Time Accurate Schemes
  - Accurately Follow Unsteady Time Scales
- Time Marching to Steady State
  - Relaxation Schemes
  - Time Like Marching
  - Multi-grid
  - Direct Methods
  - Any Trick in the Book to Get Answer Efficiently

Examples for General OΔE's are:

$$u_{n+1} = u_n + hu'_n \quad \textit{Euler Explicit}$$

$$u_{n+1} = u_n + hu'_{n+1} \quad \textit{Euler Implicit}$$

and

$$\begin{aligned} \tilde{u}_{n+1} &= u_n + hu'_n \\ u_{n+1} &= \frac{1}{2}[u_n + \tilde{u}_{n+1} + h\tilde{u}'_{n+1}] \quad \textit{Predictor Corrector} \end{aligned}$$

Stability

- Characterize Stability in the ODE Sense:
  - ODE $\rightarrow e^{\lambda t}$ where $Real(\lambda) \leq 0$
  - Characteristic eigenvalue of PDE to ODE transformation, λ
- Characterize OΔE Stability in terms of $\sigma - \lambda$ Relation
  - Amplification eigenvalue of ordinary difference equation: σ
  - For example, σ is the root of the Characteristic Equation:

1.	$\sigma - 1 - \lambda h = 0$	Explicit Euler
2.	$\sigma^2 - 2\lambda h \sigma - 1 = 0$	Leapfrog
3.	$\sigma^2 - (1 + \frac{3}{2}\lambda h)\sigma + \frac{1}{2}\lambda h = 0$	AB2
4.	$\sigma^(1 - \lambda h) - 1 = 0$	Implicit Euler
5.	$\sigma - 1 - \lambda h - \frac{1}{2}\lambda^2 h^2 - \frac{1}{6}\lambda^3 h^3 - \frac{1}{24}\lambda^4 h^4 = 0$	RK4

Table 1. Some $\lambda - \sigma$ Relations

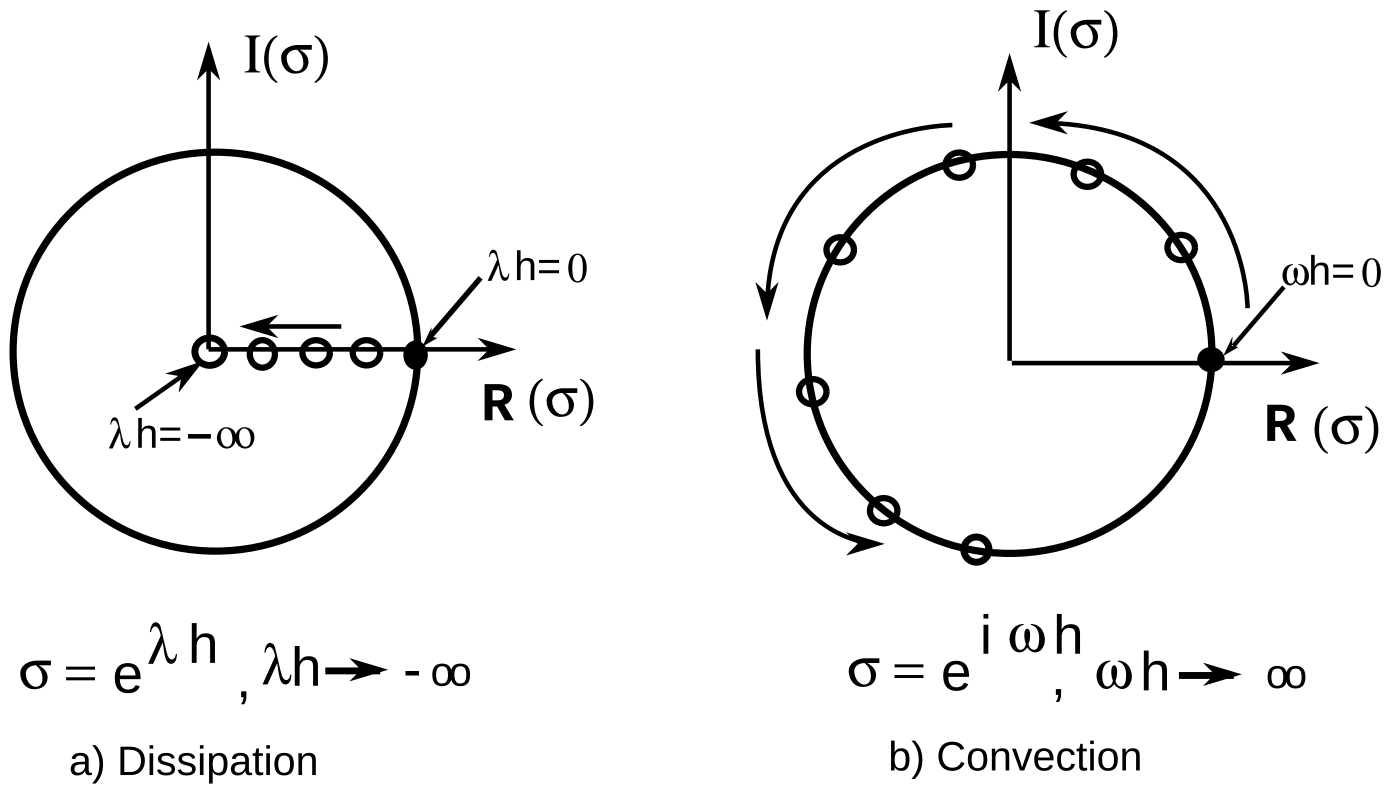
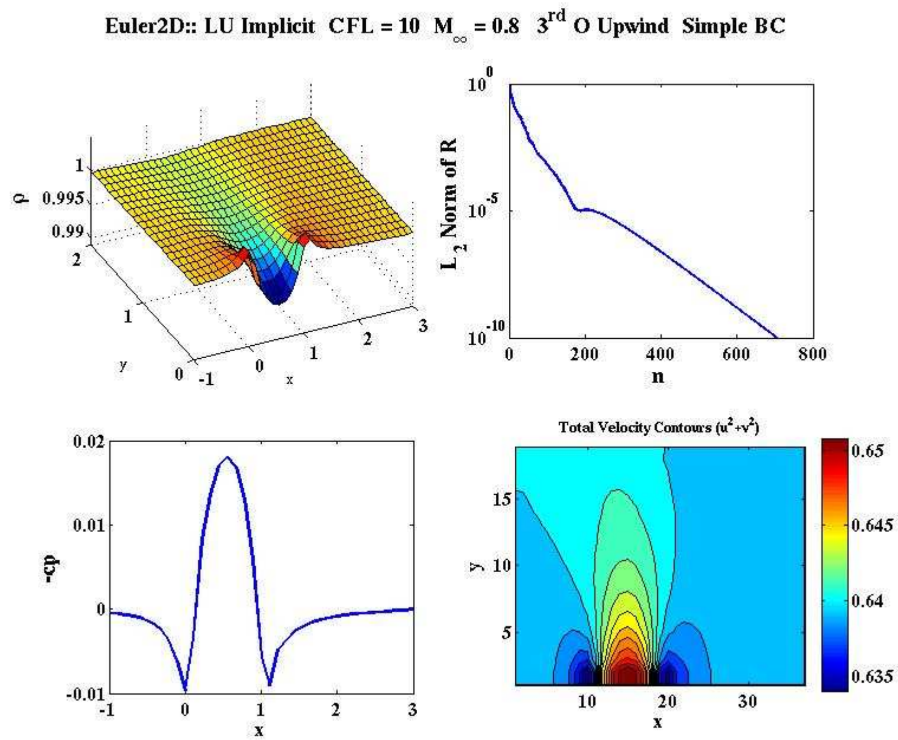


Figure 5: Exact traces of σ-roots for model equations.

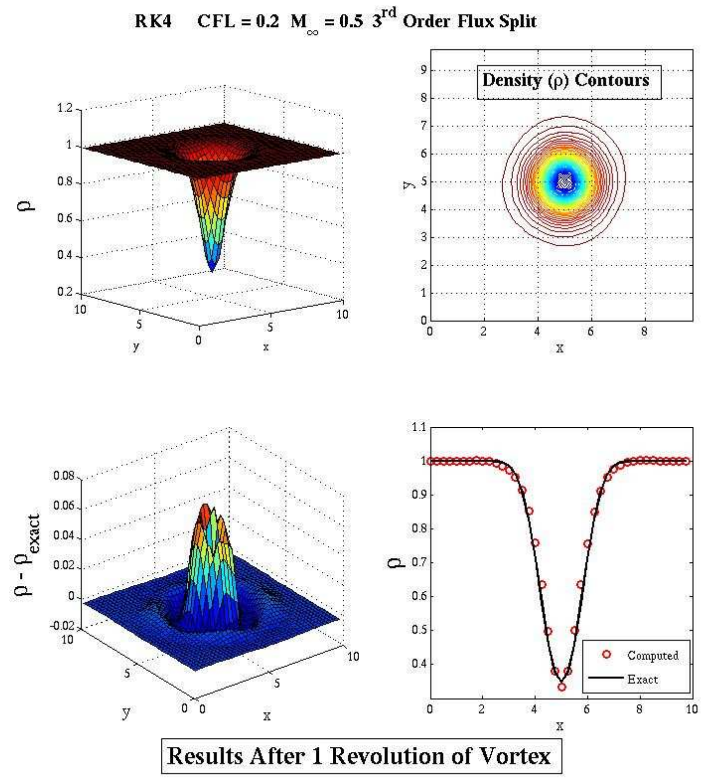
CLASS GOALS

- Apply design and analysis techniques to the Nonlinear Euler Eqs.
- Employ model equations to develop concepts
  - Convection-Diffusion Eq. $\frac{du}{dt} + a\frac{du}{dx} = \mu\frac{d^2u}{dx^2}$
- Truncation Error Analysis
  - Classical Truncation Error $O(\Delta x^p)$
  - Modified Wave Number Analysis:
    $ik^* = isin(k\Delta x)/\Delta x = ik + O(\Delta x^p) \approx ik$
- Stability Analysis, Convergence Characteristics
- Multi-Dimensional Techniques
- Nonlinear Equation Techniques: Flux Vector Splitting
- Implementation Issues: Efficiency, Parallization

Linearized Biconvex Airfoil



Nonlinear Vortex Propagation



Homework Assignments

- Assignment 1, Lab 1: Stability Examples

```
%                        Lab Assignment #1
%
%    Program the following Matlab code. The best way is to make a
%    file (filename: Lab_HW1.m) of the commands as
%    they are shown below.
%    Startup matlab on your system and at the matlab prompt
%    (typically >) type: Lab_HW1
%    The program should start up and prompt you for input.
%
%    The system you are solving is
%        the linear wave equation:
%
```

```
%          du/dt + du/dx = 0 for 0 <= x < = 2 pi
%          with periodic boundary conditions  u(0) = u(2 pi)
%
%       using
%
%                n+1    n          n       n      n
%               u    = u  - CFL(a u   + b u  + c u   )
%                j      j          j-1     j      j+1
%
%               u     = u   enforces the periodic BC
%                jmax    1
%
%   We have chosen a three point differencing in space and
%   explicit Euler time differencing.
%
%   Problem #1:  For nmax = 10, CFL = 1.0, run the program with
%      a)  a = -0.5, b = 0.0, c = 0.5 and describe what happens.
```

```
%           Short descriptions please.
%           Hand sketch the 5th and 10th curve plot of u vrs x
%           or if you have printer capability print the results.
%       b)  Repeat a) for a = -1, b = 1, c = 0
%       c)  Repeat a) for CFL = 0.7
%       d)  Repeat a) for CFL = 1.4
%
%    Problem #2:  For nmax = 10, a = -1, b = 1, c = 0
%                 run the program with
%       a)  CFL = 0.7 and describe what happens.
%           Hand sketch the 5th and 10th curve plot of u vrs x
%           or if you have printer capability print the results.
%       b)  Repeat a) CFL = 1.4
%
%    Problem #3:
%           For nmax = 10, CFL=1.0, run the program with
%       a)  a = -0.9, b = 0.8, c = 0.1 and describe what happens.
```

```
%          Short descriptions please.
%          Hand sketch the 5th and 10th curve plot of u vrs x
%          or if you have printer capability print the results.
%      b)  Repeat a) for a = -0.99, b = 0.98, c = 0.01
```